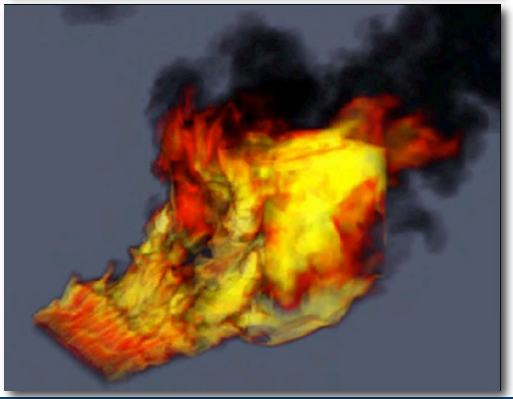


# Making Representative Apps Represent Apps

Mike Heroux

2015 Workshop on Representative Applications

Sep 8, 2015

Sandia National Laboratories

*Exceptional service in the national interest*

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

# Outline

Sandia National Laboratories

- Motivation and Background
- Miniapps Overview
- Howto on Miniapps
- Mantevo Project & Portal
- HPCG Benchmark

Sandia National Laboratories

# *MOTIVATION & BACKGROUND*
# *WHY REPRESENTATIVE APPS*

# A Confluence of Trends

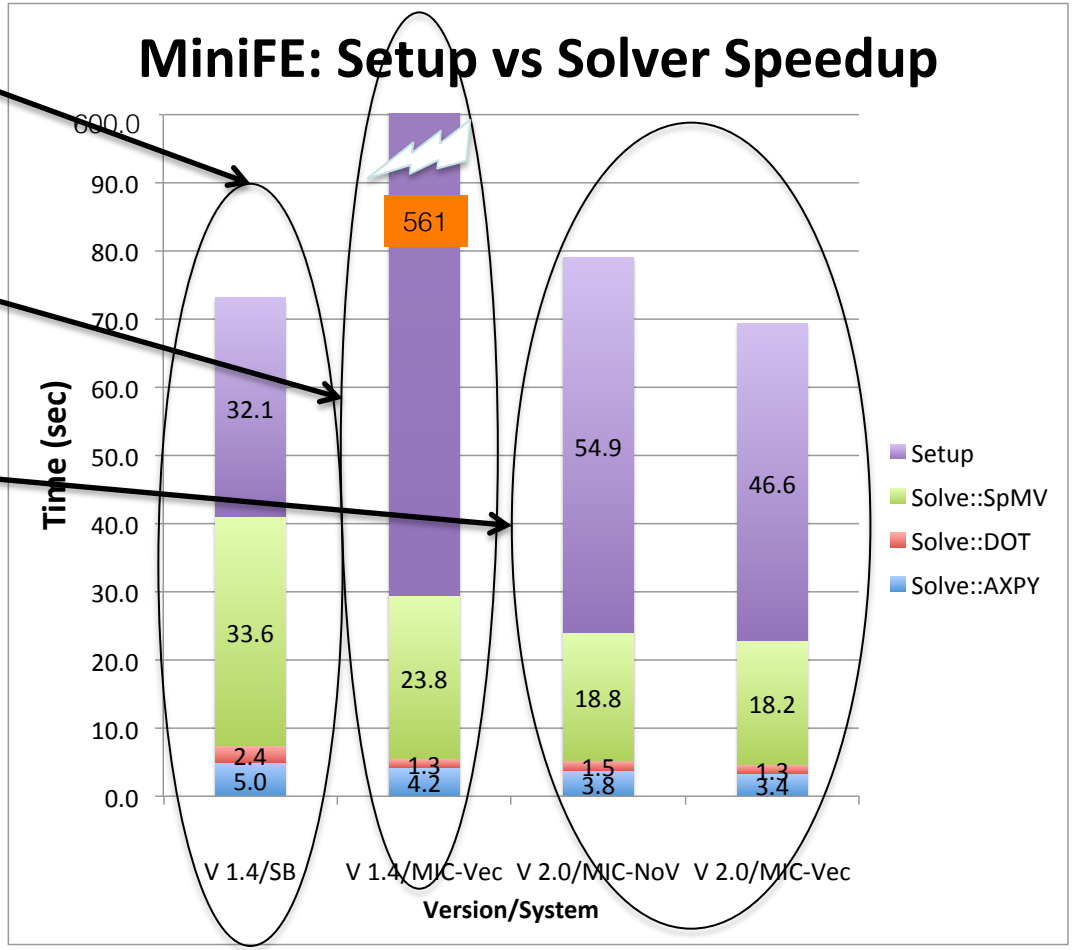
Sandia National Laboratories

- Fundamental trends:
  - Disruptive HW changes: Requires thorough alg/code refactoring.
  - Demands for coupling: Multiphysics, multiscale, pipelines.
- Challenges:
  - Need 2 refactorings: 1+ε, not 2-ε. Really: Continuous change.
  - Modest app development funding: No monolithic apps.
  - Requirements are unfolding, evolving, not fully known *a priori*.
- Opportunities:
  - Better design and SW practices & tools are available.
  - Better SW architectures: Toolkits, libraries, frameworks.
- >>>>> Co-design can be very effective.

# The work ahead of us: Threads and vectors

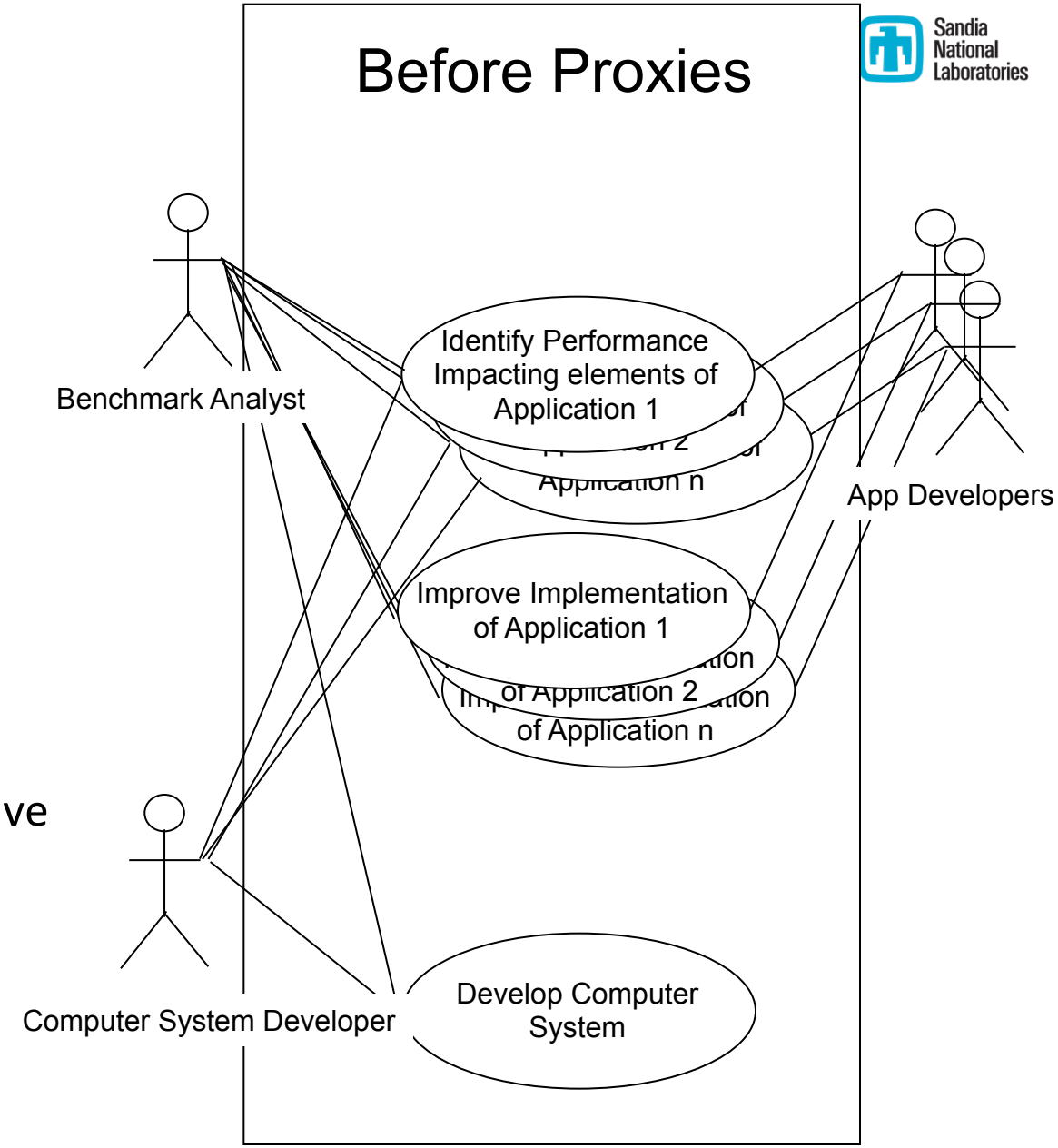
## MiniFE 1.4 vs 2.0 as Harbingers

- Typical MPI-only run:
  - Balanced setup vs solve
- First MIC run:
  - Thread/vector solver
  - No-thread setup
- V 2.0: Thread/vector
  - Lots of work:
    - Data placement, const /restrict declarations, avoid shared writes, find race conditions, …
  - Unique to each app



# Background

- Goal: Develop scalable computing capabilities via:
  - Application analysis.
  - Application improvement.
  - Computer system design.
- Fixed timeline.
- Countless design decisions.
- Collaborative effort.
- Pre-Proxies:
  - Work with each, large application.
  - Application developers have conflicting demands:
    - Features,
    - performance.
  - Application performance profiles have similarities.

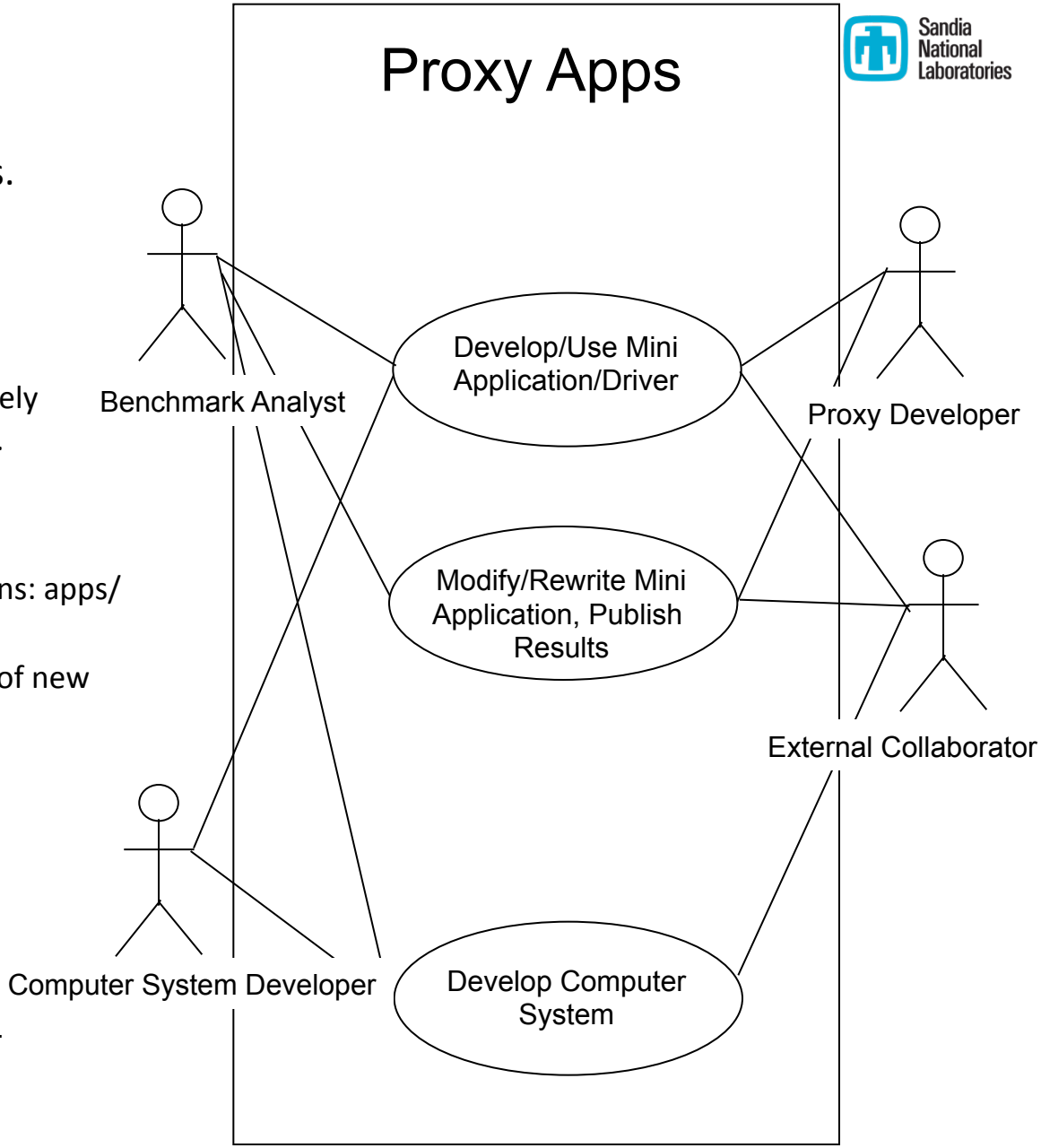


# Proxy App Effort

- Develop:
  - Proxies: Miniapps, Minidrivers.
- Goals:
  - Aid in system design decisions:
    - Proxies for real apps.
    - Easy to use, modify or completely rewrite, e.g., multicore studies.
  - Guide application and library developers:
    - Get first results in new situations: apps/ libs know what to expect.
    - Better algorithms: Exploration of new approaches.
  - Predict performance of real applications in new situations.
  - New collaborations.

## Results:

- Better-informed design decision.
- Broad dissemination of optimization techniques.
- Incorporation of external R&D results.
- **Starts *upstream* from work with real apps.**
- **Does not replace work with real apps.**



# A Listing of Application Proxies

Sandia National Laboratories

- Proxy/Representative App:
  - Generic term for all types.
- Skeleton App:
  - Communication accurate, computation fake.
- Compact App:
  - A small version of a real app.
  - Attempting some tie to physics.
- Scalable Synthetic Compact Applications (SSCA):
  - DARPA HPCS.
  - Formal specification.
  - Code and detailed spec to allow re-write.

# App Proxies (cont).

- HPC Challenge Benchmarks.
- NAS Parallel Benchmarks.
- SPEC.
- HPL: Really?
  - Yes: In the ’80s
  - Approximated:
    - Frontal solver, NASTRAN, ANSYS, more.
    - Multifrontal/Supernodal solver: First Gordon Bell.
  - Question: Why are DCA++, LSMS fastest apps?
  - Answer: HPL was first co-design vehicle.

# … And More: A crowded space

Sandia National Laboratories

- UHPC Challenge Problems:
  - Formal specification.
  - Math, kernel extraction.
  - Intended to be open source?
- Motifs, aka dwarves.
  - Really are patterns, not actionable.

  "Even as cartoon characters they are sketchy."

  (John Lewis)
- Miniapps:
  - Recent addition to a much larger eco-system.
  - Twist: Balance of size, cohesive performance coupling, freedom to re-write
  - Focus: Insight, information.

# Everything is an Application Proxy

## But there are different kinds

***ASC Co-design Proxy App Strategy***

Mike Heroux (maherou@sandia.gov)
Rob Neely (neely4@llnl.gov)
Sriram Swaminarayan (sriram@lanl.gov)
Version 1.0: 2013-02-12

- **Kernels**: Kernels are one or more small code fragments or data layouts that are used extensively by the applications and are deemed essential to optimal performance on next generation advanced systems. They are useful for testing programming methods and performance at the node level, and typically do not involve network communication (MPI). Their small size also makes them ideal for doing early evaluation and explorations on hardware emulators and simulators. A kernel is a standalone piece of code that is small and performance- and tradeoff-impacting, even though decoupled from other application components.
- **Skeleton apps**: Skeleton apps reproduce the memory or communication patterns of a physics application or package, and make little or no attempt to investigate numerical performance. They are useful for targeted investigations such as network performance characteristics at large scale, memory access patterns, thread overheads, bus transfer overheads, system software requirements, I/O patterns, and new programming models. Skeleton also may allow the release of more applications as non-export controlled by removing mathematical or algorithmic details while still conveying useful performance information.
- **Mini apps**: Mini apps combine some or all of the dominant numerical kernels contained in an actual stand-alone application and produce simplifications of physical phenomena. This category may also include libraries wrapped in a test driver providing representative inputs. They may also be hard-coded to solve a particular test case so as to simplify the need for parsing input files and mesh descriptions. Mini apps range in scale from partial, performance-coupled components of the application to a simplified representation of a complete execution path through the application.
- The definition of *mini apps* now includes those that were previously called *compact apps*. We combined these categories to make communication within the community easier, as the distinction was not easy to define to a broad audience.

Sandia National Laboratories

# *MINIAPPS*

# Miniapps : Tools enabling exploration

Sandia National Laboratories

| Focus | Proxy for a key app performance issue |
|---|---|
| Intent | Tool for codesign: output is information |
| Scope of change | Any and all |
| Size | A few thousand lines of code |
| Availability | Open source (LGPL) |
| Developer/owner | Application team |
| Life span | Until its no longer useful |

***Related:***

| Benchmark | Output: metric to be ranked. |
|---|---|
| Compact app | Application relevant answer. |
| Skeleton app | Inter-process comm, application “fake” |
| Proxy app | Über notion – Representative App |

Sandia National Laboratories

# *MINIAPP HOWTO*

# Miniapp Howto Overview

Sandia National Laboratories

- Ingredients:
  - People: Development roles.
  - Abstract Machine and Execution Model.
  - Constraints on size and portability.
  - Simple build, control, data capturing.
- Steps:
  - Go/No-go.
  - Brainstorming.
  - Base Implementation.
  - Validation.
  - Design studies.
  - Retirement.

# People

Sandia National Laboratories

- Application development experts. Know:
  - Range of algorithms.
  - The targeted full-scale app.
- Parallel patterns experts. Know:
  - A taxonomy of parallel patterns.
    - Parallel-for, parallel-reduce, task-graph, …
  - Parallel execution strategies.
- Architecture/system experts. Know:
  - Architecture trends.
  - Performance/cost tradeoffs.

# Howto Step 0

Sandia National Laboratories

Step 0: Go/No-go.

- Small, portable app may need no miniapp:
  - 50K-100K lines.
  - Few 3rd party lib dependencies.
  - Small problem domain.
  - Modular (easy to scope down).
- No need for miniapp.

# Charon Complexity

Sandia National Laboratories

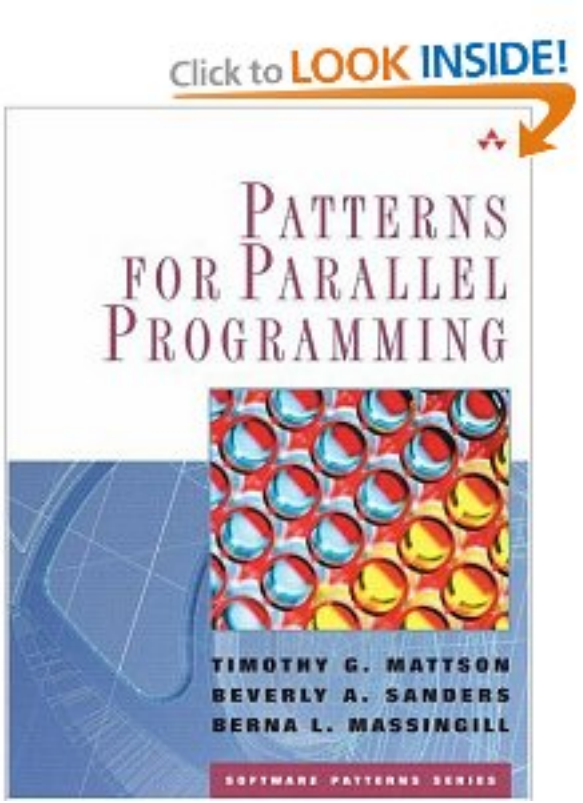
- SLOCCOUNT (tool from David A. Wheeler).
  - Charon physics: 191,877 SLOC.
  - Charon + nevada framework 414,885 SLOC
  - Charon_TPL 4,022,296 SLOC
- Library dependencies:
  - 25 Trilinos package.
  - 15 other TPLs.
- Requires “heroic effort” to build.
- MPI-only, no intranode parallelism.
- Export controlled.
- Stats courtesy of Roger Pawlowski.

# Howto Step 1

Sandia National Laboratories

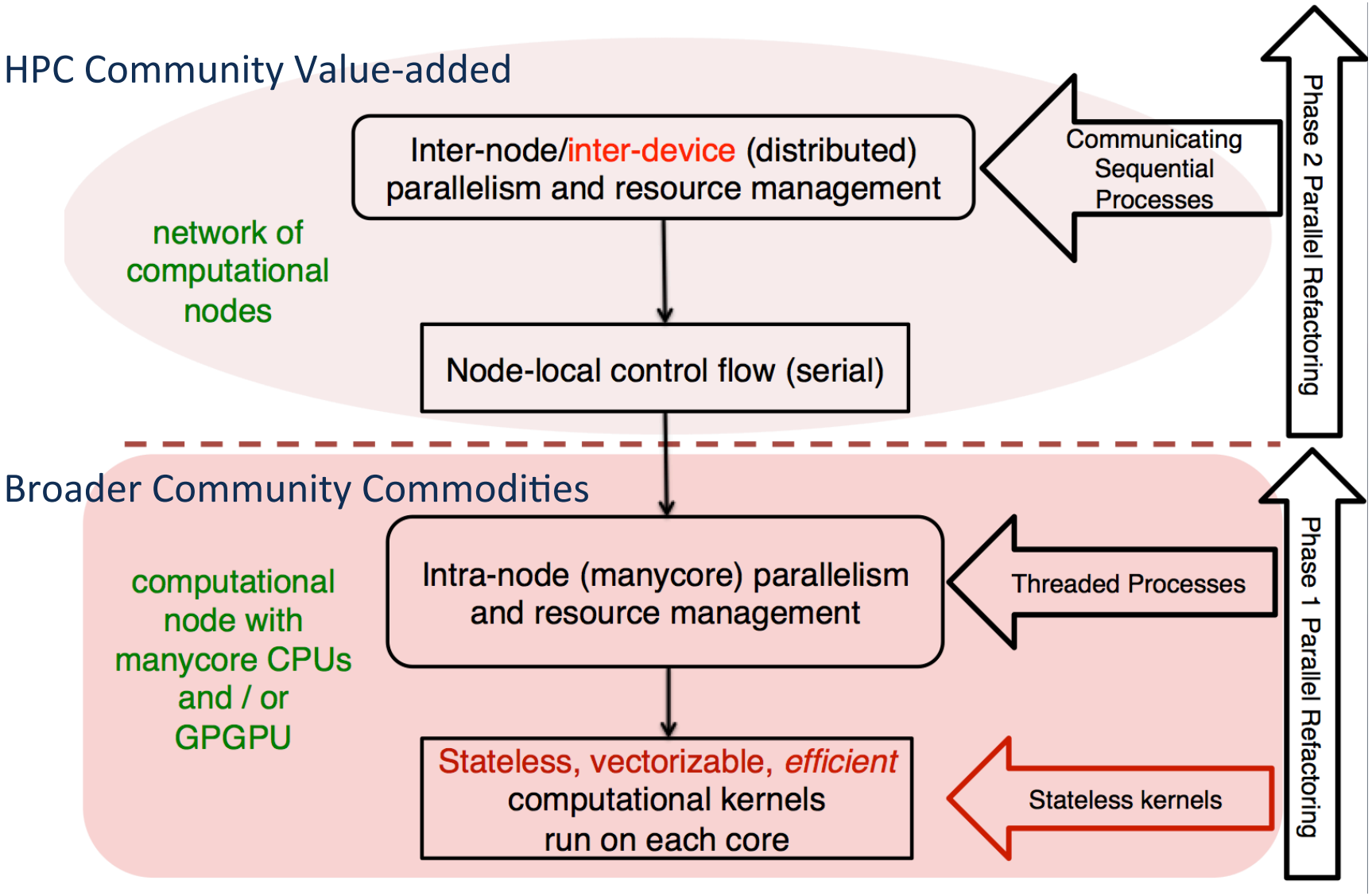
Step 1: Brainstorming:

- Pick a design point (or two).
- ID parallel patterns:
  - MiniFE:
    - SPMD, cache blocking, vectorization, pipeline, task-graph
    - CG solver: SpMV, axpy, dot.
- Step 1a: Design:
  - Organize miniapp in terms of patterns.



# MPI+X Abstract Machine Architecture

Sandia National Laboratories



# Howto Step 2

Sandia National Laboratories

Step 2: Base Implementation:

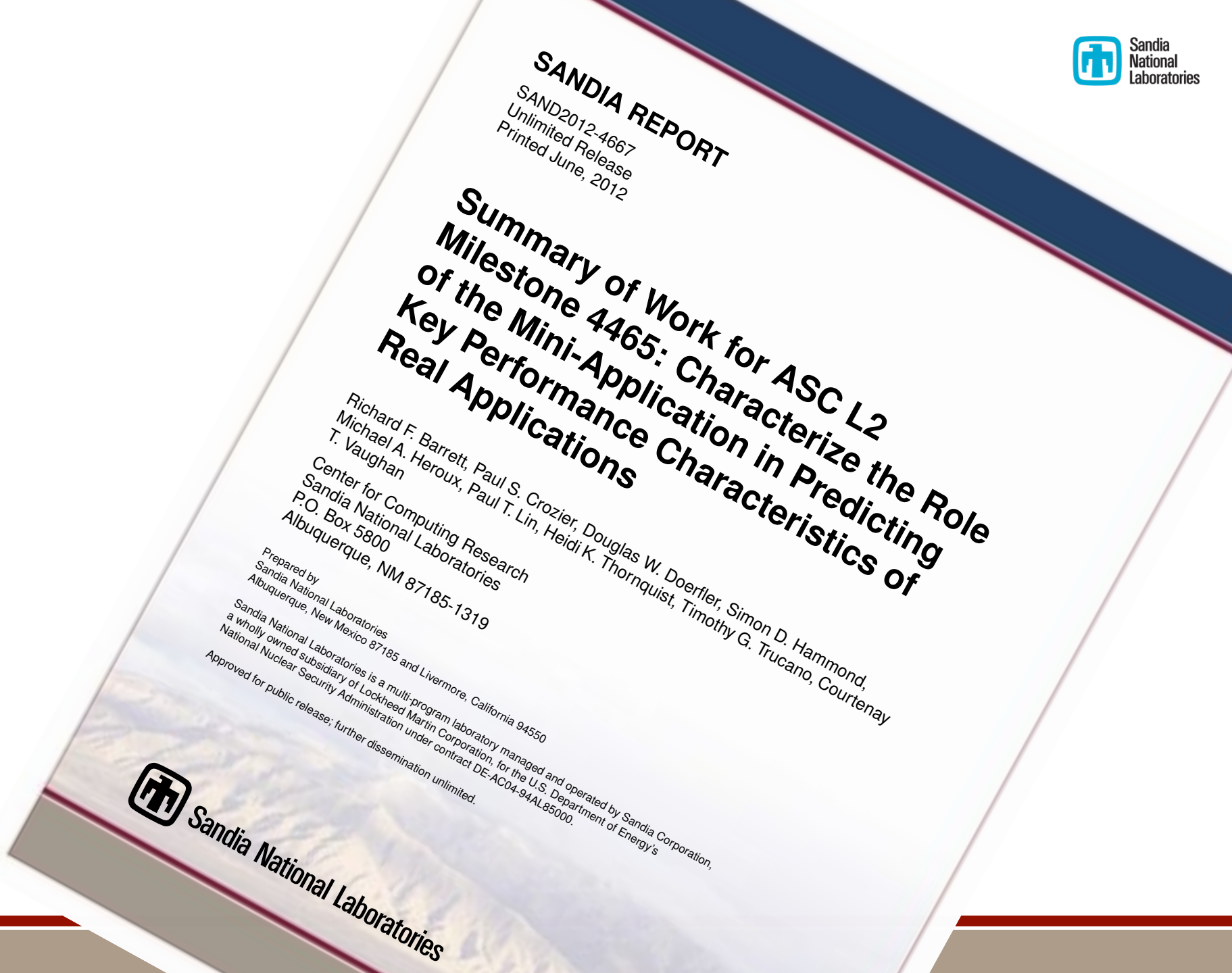
- MPI+OpenMP:
- Four basic build:
  - Serial, OpenMP, MPI, MPI+OpenMP.
- Hand-coded makefiles.
- A “few thousand” lines of code (loose metric).
- Derivatives should not pollute base.
  - Derived versions should be kept separate.

# Howto Step 3

Sandia National Laboratories

Step 3: Validation:

- Lots of experiments: Compare mini and target app.
- Proxy is a performance *model* for real app.
- Learn what it models well or poorly.


SANDIA REPORT
SAND2012-4667
Unlimited Release
Printed June, 2012
Summary of Work for ASC L2 Milestone 4465: Characterize the Role of the Mini-Application in Predicting Key Performance Characteristics of Real Applications
Richard F. Barrett, Paul S. Crozier, Douglas W. Doerfler, Simon D. Hammond, Michael A. Heroux, Paul T. Lin, Heidi K. Thornquist, Timothy G. Trucano, Courtenay T. Vaughan
Center for Computing Research
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1319
Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.
Approved for public release; further dissemination unlimited.
Sandia National Laboratories

# Howto Step 4

Sandia National Laboratories

Step 4: Design studies.

- The real fun.
- Design space exploration.
- What if…
- Collaborate: Users from all over.
- This is the fun stuff:
  - Exploring design options.
  - Getting insight.
  - Making better-informed decisions.
  - Far upstream from when real apps are feasible.

# A few known impacts

Sandia National Laboratories

- Design of Linear Alegbra kernels for the Kokkos LinAlg and Teptra stacks that came from work on MiniFE in-house and with NVIDIA and Intel.
  - The use of MiniFE as a benchmark and a design exploration vehicle here helped us improve performance, code readability and inspired Intel to look at new cost functions in the compiler toolchain. The knowledge gained from these studies is now being added into Trilinos for the upcoming release.
- Christian Trott's work with MiniMD as a design vehicle, again both in benchmark and in programming model research.
  - Has had a huge impact on features in Kokkos and is now being wound into LAMMPS for their forthcoming releases.
- MiniFE and MiniMD pushing changes into how OpenACC relates to C++ objects.
  - Driving changes at the specification level not just in compilers.
- Courtenay Vaughan's work with MiniAMR is being used to drive communication pattern research with IBM, Cray and Intel.
  - Both from a "run the app" perspective but also from an extract and simulate approach.
- MiniGhost studies from Richard Barrett and Courtenay.
  - Reordered MPI ranks to improve code performance, now added to CTH.

# Step 4a: Data Management Common Look-and-Feel: YAML

Sandia National Laboratories

- Input parameters:
  - Command line.
  - YAML file.
- Output:
  - YAML.
  - Embeds input parameters.
  - Output file can be input.
- Data parsing and collection:
  - Email list submission of YAML file.
  - CoPylot: Digests email, populates database.
- Common YAML data functions across all miniapps.

YAML ain't a Markup Language
- *de facto* standard format
- Human readable
- Convertible to/from XML, others

```
currentElement->get("performance_summary")->add("total","");
currentElement->get("performance_summary")->get("total")->add("time",times[0]);
currentElement->get("performance_summary")->get("total")->add("flops",3.0*fnops);
currentElement->get("performance_summary")->get("total")->add("mflops",3.0*fnops/times[0]/1.0E6);
```

# YAML Output File Excerpts

```
beefy.109% ./miniFE.x nx=30 ny=30 nz=30
 creating/filling mesh...0.00031209s, total time: 0.00031209
generating matrix structure...0.0196991s, total time: 0.0200112
       assembling FE data...
get-nodes: 0.0035727
compute-elems: 0.090822
sum-in: 0.0277233
0.125864s, total time: 0.145875
     imposing Dirichlet BC...0.0176551s, total time: 0.16353
making matrix indices local...8.10623e-06s, total time: 0.163538
Starting CG solver ...
Initial Residual = 182.699
Iteration = 5   Residual = 43.6016
Iteration = 10   Residual = 6.13924
Iteration = 15   Residual = 0.949901
Iteration = 20   Residual = 0.131992
Iteration = 25   Residual = 0.0196088

...
```

```
Platform:
  hostname: beefy.cs.csbsju.edu
  kernel name: 'Linux'
  kernel release: '2.6.34.7-66.fc13.x86_64'
  processor: 'x86_64'
Build:
  CXX: '/usr/lib64/openmpi/bin/mpicxx'
  compiler version: 'g++ (GCC) 4.4.5 20101112 (Red Hat 4.4.5-2)'
  CXXFLAGS: '-O3'
  using MPI: yes
  Threading: none
Run Date/Time: 2011-03-14, 22-30-26
Rows-per-proc Load Imbalance:
  Largest (from avg, %): 0
  Std Dev (%): 0
…

Total:
    Total CG Time: 0.065695
    Total CG Flops: 9.45762e+07
    Total CG Mflops: 1439.63
  Time per iteration: 0.0013139
Total Program Time: 0.237604
```

# Howto Step 5

Sandia National Laboratories

Step 5: Retirement.

- Need to avoid HPL, NAS Benchmarks entrenchment.
- Example: HPCCG was first miniapp.
  - Seldom used today.
  - However, it is a go-to still for early studies, e.g., PIM.
  - Retirement may be a cyclic process.

# Miniapp Howto Summary

Sandia National Laboratories

- Ingredients:
  - People: Development roles.
  - Abstract Machine and Execution Model.
  - Constraints on size and portability.
  - Simple build, control, data capturing.
- Steps:
  - Go/No-go.
  - Brainstorming.
  - Base Implementation.
  - Validation.
  - Design studies & collaboration.
  - Retirement.

Sandia National Laboratories

# *MANTEVO PROJECT & PORTAL*

# Mantevo* Project

- Multi-faceted performance project.
- Started 8 years ago.
- Two types of packages:
  - Miniapps: Small, self-contained programs.
    - MiniFE/HPCCG: unstructured implicit FEM/FVM.
    - phdMesh: explicit FEM, contact detection.
    - MiniMD: MD Force computations.
    - MiniXyce: Circuit RC ladder.
    - MiniGhost: Data exchange pattern of CTH.
  - Minidrivers: Wrappers around Trilinos packages.
    - Beam: Intrepid+FEI+Trilinos solvers.
    - Epetra Benchmark Tests: Core Epetra kernels.
- Open Source (LGPL)
- Staffing: Application & Library developers.




* Greek: augur, guess, predict, presage

# Mantevo 1.0 (SC12)

Sandia National Laboratories

| miniApp | |
|---|---|
| CloverLeaf | Compressible Euler eqns, explicit 2nd order accurate |
| CoMD | Molecular dynamics (SPaSM) |
| HPCCG | Unstructured implicit finite element |
| miniFE | Implicit finite element solver |
| miniGhost | FDM/FVM explicit (halo exchange focus) |
| miniMD | Molecular dynamics (Lennard-Jones) |
| miniXyce | SPICE-style circuit simulator |
| mini"Aero" | *In development* |
| miniAMR | Adaptive mesh refinement of an Eulerian mesh |

*1.0 release*

Mantevo releases annually.
Just prior to Supercomputing.

# Mantevo 2.0 (SC13)

Sandia National Laboratories

| miniApp or miniDriver | | |
|---|---|---|
| Cleverleaf 1.0 | Eulerian on structured grid with AMR | |
| CloverLeaf 1.0.1 | Compressible Euler eqns, explicit 2nd order accurate | |
| CoMD 1.1 | Molecular dynamics (SPaSM) | |
| EpetraBenchmarkTest 1.0 | Exercises Epetra sparse and dense kernels. | |
| HPCCG 1.0 | Unstructured implicit finite element | *2.0 release* |
| miniFE 2.0 | Implicit finite element solver | |
| miniGhost 1.0.1 | FDM/FVM explicit (halo exchange focus) | |
| miniMD 1.2 | Molecular dynamics (Lennard-Jones) | |
| miniXyce 1.0 | SPICE-style circuit simulator | |
| mini"Aero"* | *In development* | |
| miniAMR | Adaptive mesh refinement of an Eulerian mesh | |

# Mantevo 3.0 (SC14)

Sandia National Laboratories

| miniApp or miniDriver | | |
|---|---|---|
| **Cleverleaf 2.0 (up from 1.0)** | Eulerian on structured grid with AMR | |
| **CloverLeaf 1.1 (up from 1.0.1)** | Compressible Euler eqns, explicit 2nd order accurate | |
| CoMD 1.1 | Molecular dynamics (SPaSM) | |
| EpetraBenchmarkTest 1.0 | Exercises Epetra sparse and dense kernels. | |
| HPCCG 1.0 | Unstructured implicit finite element | *2.0 release* |
| **miniFE 2.0.1 (up from 2.0)** | Implicit finite element solver | |
| miniGhost 1.0.1 | FDM/FVM explicit (halo exchange focus) | |
| miniMD 1.2 | Molecular dynamics (Lennard-Jones) | |
| miniXyce 1.0 | SPICE-style circuit simulator | |
| miniAMR 1.0 | Adaptive mesh refinement of an Eulerian mesh | |
| miniSMAC2D 2.0 | FD 2D incompressible N/S on a structured grid. | + |
| PathFinder 1.0 | Signature search | |
| miniAero 1.0 | 3D unstr FV R-K 4th order time, inviscid Roe Flux | |
| TeaLeaf 1.0 | Solid mechanics | |

# mantevo.org Website

Sandia National Laboratories

- Single portal for project:
  - Access to downloads.
  - Developer tools.
- Downloads:
  - Tarballs from Mantevo Repository.
  - Links to externally-available packages (AWE efforts).
- Developer tools:
  - New code proposal checklist.
  - Release checklist, includes Mantevo “common look-and-feel” requirements.
    - Use simple makefiles, output in YAML.
    - Reference implementation requirements.
    - Etc.

# General Comments

Sandia National Laboratories

- Common Look-and-feel choice are important:
  - Simple Makefile:
    - Really important in brand new environment (vs. cmake for example).
  - YAML:
    - Readable text output.
    - Convertible to data records.
  - Serial, OpenMP, MPI, MPI+OpenMP versions:
    - Serial is very important (when no MPI compiler/libs).
    - OpenMP can help, but only if scalable thread refactoring done.
- Persistent concerns:
  - Validation of miniapps as performance proxies.
  - Use as benchmark (OK if done carefully).
  - Need for end-to-end workflow proxies.

Sandia National Laboratories

# *A DIFFERENT REPRESENTATIVE APP: THE HPCG BENCHMARK*

Sandia National Laboratories

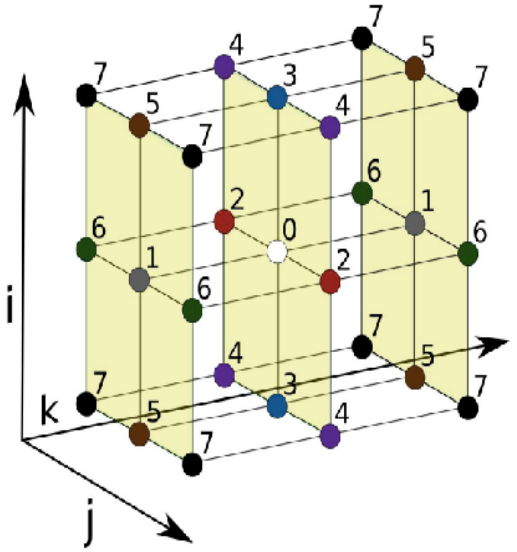
# HPCG Snapshot

- High Performance Conjugate Gradient (HPCG).
- Solves $Ax=b$, $A$ large, sparse, $b$ known, $x$ computed.
- An optimized implementation of PCG contains essential computational and communication patterns that are prevalent in a variety of methods for discretization and numerical solution of PDEs

- Patterns:
  - Dense and sparse computations.
  - Dense and sparse collective.
  - Multi-scale execution of kernels via MG (truncated) V cycle.
  - Data-driven parallelism (unstructured sparse triangular solves).
- Strong verification (via spectral properties of PCG).

# Model Problem Description

- Synthetic discretized 3D PDE (FEM, FVM, FDM).
- Zero Dirichlet BCs, Synthetic RHS s.t. solution = 1.
- Local domain: $(n_x \times n_y \times n_z)$
- Process layout: $(np_x \times np_y \times np_z)$
- Global domain: $(n_x * np_x) \times (n_y * np_y) \times (n_z * np_z)$
- Sparse matrix:
  - 27 nonzeros/row interior.
  - 8 – 18 on boundary.
  - Symmetric positive definite.



27-point stencil operator

# Merits of HPCG

Sandia National Laboratories

- Includes major communication/computational patterns.
  - Represents a minimal collection of the major patterns.
- Rewards investment in:
  - High-performance collective ops.
  - Local memory system performance.
  - Low latency cooperative threading.
- Detects/measures variances from bitwise reproducibility.
- Executes kernels at several (tunable) granularities:
  - nx = ny = nz = 104 gives
  - nlocal = 1,124,864; 140,608; 17,576; 2,197
  - ComputeSymGS with multicoloring adds one more level:
    - 8 colors.
    - Average size of color = 275.
    - Size ratio (largest:smallest): 4096
  - Provide a “natural” incentive to run a big problem.

# HPL vs. HPCG: Bookends

Sandia National Laboratories

- Some see HPL and HPCG as “bookends” of a spectrum.
  - Applications teams know where their codes lie on the spectrum.
  - Can gauge performance on a system using both HPL and HPCG numbers.

Sandia National Laboratories

# HPCG STATUS

# Special Issue: International Journal of High Performance Computer Applications

Sandia National Laboratories

1. Reference HPCG.
2. Intel.
3. Nvidia.
4. NUDT.
5. Riken.
6. Coming a little later: IBM.

- Discussion and results from vendor optimizations.
- Some articles are available, others in final review.
- Some highlights…

# Rewards investment high performance collectives.

Sandia National Laboratories

"Edison spends only 1.9% of the total time in all-reduce while SuperMUC, Occigen, and Stampede spend 12.9%, 5.9%, and 22.0%, respectively. We believe this difference primarily comes from that Edison uses a low-diameter high-radix Aries network with Dragonfly topology."

Intel HPCG Paper

# Collectives futures

- “Addressing the bottleneck in collective communications will be also an important challenge as the collectives are shown to often take well above 10% of the total time. Even though high-radix Dragonfly topology considerably speedups the collectives, we envision that continued innovation in network infrastructure will be necessary due to ever increasing concurrency in high performance computing systems.”

# Impact broader set of computations

Sandia National Laboratories

"The optimizations described in this paper are not limited to the HPCG benchmark and can be also applicable to other problems and sparse solvers as exemplified by our evaluation with unstructured matrices shown in [our previous report]."

# Looking toward next generation memories

Sandia National Laboratories

"We expect challenges and opportunities laid out for HPCG in the next few years. One of the significant challenges will be effective use of emerging memory technologies and the accompanied diversification of memory hierarchy."

# Detecting FP Variations (Reproducibility)

Sandia National Laboratories

Residual=4.25079640861055785883e-08 (0x1.6d240066fda73p-25)
Residual=4.25079640861032293954e-08 (0x1.6d240066fd910p-25)
Residual=4.25079640861079079289e-08 (0x1.6d240066fdbd3p-25)
Residual=4.25079640861054528568e-08 (0x1.6d240066fda60p-25)
Residual=4.25079640861068491377e-08 (0x1.6d240066fdb33p-25)
Residual=4.25079640861059094605e-08 (0x1.6d240066fdaa5p-25)

"The code correctly identified small variations in the residuals, caused by the network off-loading collectives. There is a small improvement in performance but the off-loading collectives introduce a small non-reproducibility."

# Vendor improvement: Intel 4X

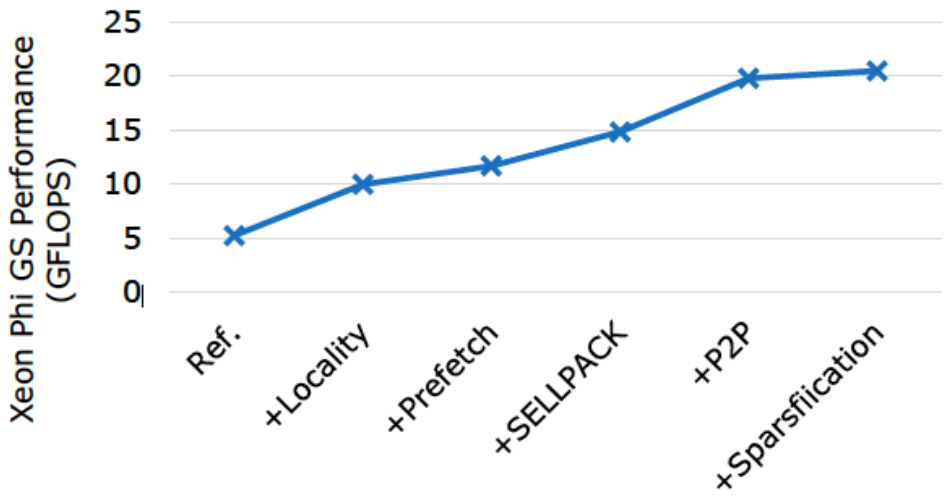



Fig. 5: The impact of optimizations on the Xeon Phi performance of SymGS parallelized with task scheduling.

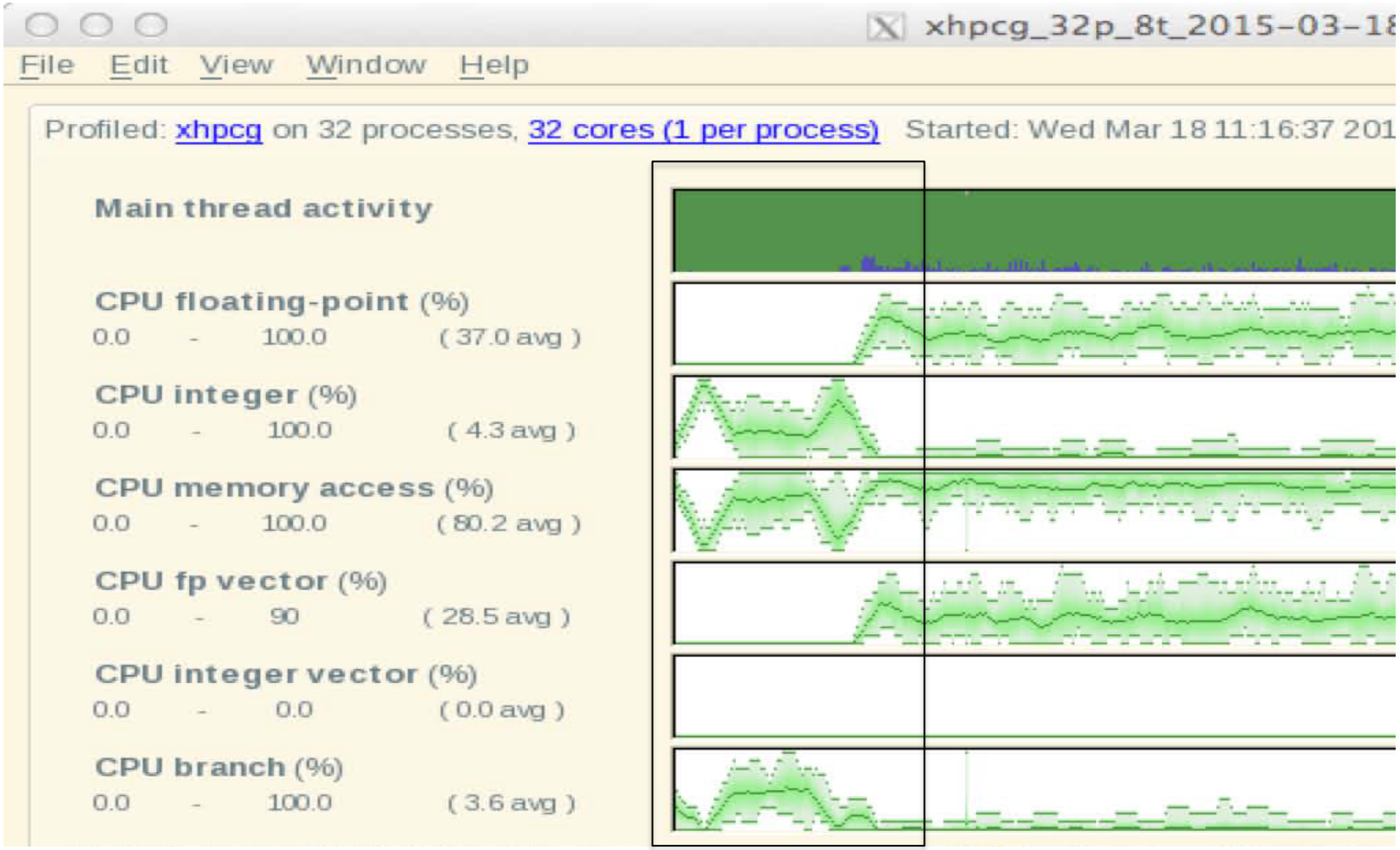
- `Ref.`: the reference implementation ran with 240 MPI ranks
- `+Locality`: storage layout optimization for locality (Section IV-A1)
- `+Prefetch`: software prefetches
- `+SELLPACK`: vectorization-friendly matrix storage format [43]
- `+P2P`: point-to-point synchronization instead of barriers
- `+Sparsification`: eliminating unnecessary synchronization [10]

# Next (and last) Major Version 3.X

Sandia National Laboratories

- Concern:  Too much like STREAMS.
  - Not true, from previous results.
  - Still: Interested in mixing in address/integer/logic instructions.
- Approach:
  - Time problem generation.
  - Include this time as part of overhead.
  - Overhead: Generation + Vendor optimization costs.

# HPCG 2.4 Profile (Allinea output)



# Other Items

Sandia National Laboratories

- Reference version on GitHub:
  - https://github.com/hpcg-benchmark/hpcg
  - Website:  hpcg-benchark.org, includes results auto-upload from yaml.
  - Mail list hpcg.benchmark@gmail.com
- Next event: SC'15:
  - 42 entries so far, expect more.
  - Release of HPCG 3.0.
  - Transition from version 2.4 to 3.0 is under discussion.

# HPCG ISC'15 Highlights

- 42 Systems:
  - Up from 25 at SC'14 and 15 at ISC'14.
  - Most entries from the very top of the TOP500 list.
- New supercomputers (also coming to TOP500) are:
  - KAUST Shaheen II
  - Moscow State: Lomonosov 2
- Strong showing from Japan and NEC SX machines:
  - Achieve over 10% of peak performance with HPCG
- Updated results from TACC with larger scale of the system tested.
- IBM BlueGene machines make their first appearance on the list.

# Summary

Sandia National Laboratories

- Representative apps are an essential co-design tool.
- There is no “one-size-fits-all” approach.
- Focus on upstream value proposition: Before real apps can be used.
- App teams should value, even own, the proxy.
- Some key proxy types:
  - Miniapp: Re-writeable, no restrictions, purest collaboration vehicle.
  - Minidriver: Proxy for app use of performance-impacting libraries.
  - Benchmark: Incentivizes vendor investment in important performance features.
- Some next steps:
  - Integration of proxy approach into app development efforts – make common.
  - Workflow characterization: Aggregate system behavior – “midi” apps.
  - Sunset strategies: How do we retire proxies?